بسمه تعالی

تمرین سری ششم درس یادگیری عمیق

پوریا محمدی نسب

۴۰۰۷۲۲۱۳۸

فهرست

سوال ۱.

با توجه به مقاله زیر بیان کنید این شیوه چگونه نسبت به بهینه سازهای رایج بهتر عمل میکند و ایده آن را توضیح دهید.(خواندن بخش ۳ و ۴ مقاله برای پاسخ گویی کافی است، اما خواندن کل مقاله جهت افزایش اطلاعات توصیه می گردد)

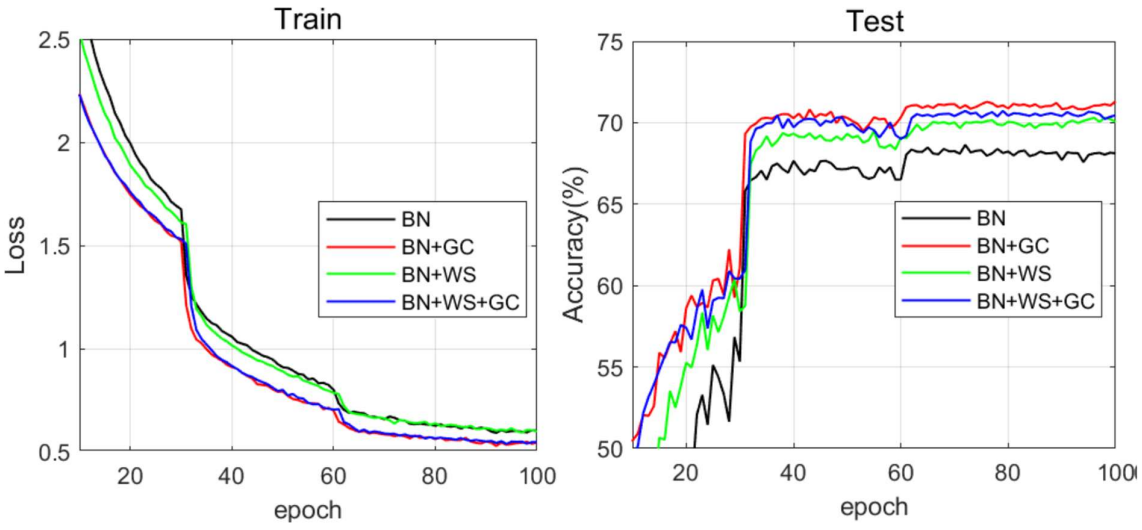
Yong, Hongwei & Huang, Jianqiang & Hua, Xiansheng & Zhang, Lei." Gradient Centralization: A New Optimization Technique for Deep Neural Networks", European Conference on Computer Vision, 2020.

هدف از این روش این است که بتوان با انجام عملیاتی روی گرادیان فرآیند آموزش را موثرتر و پایدارتر کرد. یک راه حل نرمال سازی گرادیان به کمک استاندارد سازی Z-score است که روش های BN و WS از آن استفاده کردند. متاسفانه استانداردسازی گرادیان کمکی به پایداری فرآیند آموزش نمیکند. در این مقاله روشی معرفی شده است که میانگین بردار گرادیان را محاسبه میکند و آنها را به میانگین صفر متمرکز میکند. یکی از دلایلی که این روش را مورد توجه قرار داده است ساده بودن این روش است. فرض کنید بردار گرادیان یک لایه ی کاملا متصل را با $\nabla_{wi}\mathcal{L}$ نشان دهیم. این بردار گرادیان به صورت ورودی تابع استانداردسازی گرادیان وارد میشود و خروجی برداری است که متمرکز شده با میانگین صفر است. رابطه ی زیر چگونگی انجام کار را نمایش میدهد:

$$\phi_{GC}(\nabla_{wi}\mathcal{L}) = \nabla_{wi}\mathcal{L} - \mu_{\nabla_{wi}\mathcal{L}} = \nabla_{wi}\mathcal{L} - \frac{1}{M}\sum_{j=1}^{M}\nabla_{wi,j}\mathcal{L}$$

از فواید این روش میتوان به شتاب بخشیدن به فرآیند آموزش و قابلیت تعمیم مدل اشاره کرد. افزایش قدرت تعمیم را میتوان از دو زاویه بررسی کرد: ۱) تعمیم فضای وزن ۲) تعمیم خروجی فضای ویژگی

با ترکیب این روش با روش های دیگر میتوان نتایج بهتری نیز از مدل گرفت. شکل زیر مقایسه ای بین ترکیبات مختلف روش های بهینه ساز انجام میدهد.

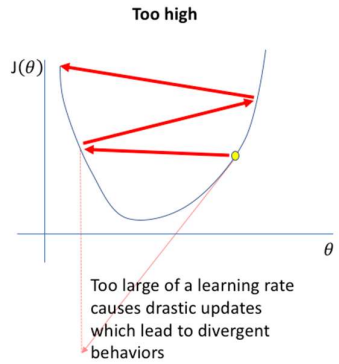


سوال ۲.

سوالات زیر را با توجه به مبحث بهینه سازها پاسخ دهید. در صورت استفاده از منابع، آنها را ذکر کنید.

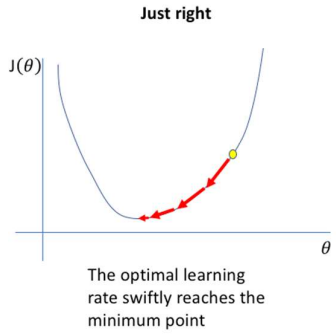
الف)استفاده از نرخ یادگیری بسیار بالا چه مشکلاتی را ایجاد میکند؟ چگونه میتوان این مشکلات را تشخیص داد؟

استفاده از نرخ آموزش بسیار بالا باعث پرش های بسیار بزرگی در فضای جستجو میشود و احتمال پریدن(رد کردن) جواب بهینه را بالا میبرد. از طرفی در برخی موارد حتی میتواند موجب واگرا شدن جواب شود. روش تشخیص این مشکل نتایج بسیار بد مدل پس از تعداد epoch های زیاد است که نشانه ای از واگرا شدن جواب دارد و یا اینکه مدل از جایی تغییری نمیکند و در عین حال به نتیجه خوبی نیز نرسیده است. شکل زیر عملکرد الگوریتم را با نرخ آموزشی بالا نمایش میدهد.

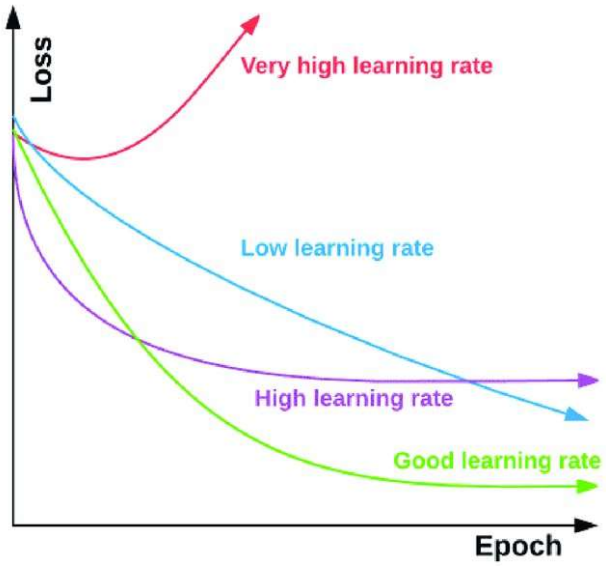


ب)استفاده از نرخ یادگیری بسیار پایین چه مشکلاتی را ایجاد میکند؟ چگونه میتوان این مشکلات را تشخیص داد؟

بر خلاف نرخ آموزش بالا، هنگامی که نرخ آموزش پایین است احتمال اینکه از جواب بهینه رد شویم بسیار کم است و همنیطور مدل واگرا نخواهد شد. اما از طرفی مشکل این است که با سرعت بسیار کمی به نقطه بهینه نزدیک میشود و البته ممکن است در نقطه بهینه محلی گیر کند و نتواند از بهینه محلی خارج شود.

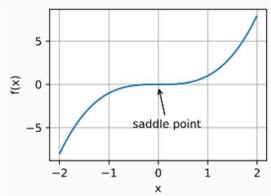


در نمودار زیر، مقایسه ای بین نرخ آموزش بالا، پایین و نرخ آموزش بهینه مشاهده میکنید.

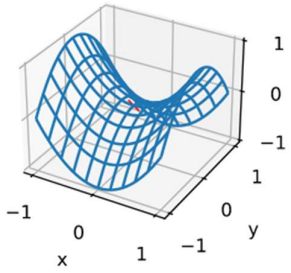


ج)نقطه زینی چیست؟ دو الگوریتم Adam و SGD را در برخورد با این نقاط مقایسه کنید. مزایا و مضرات هرکدام را بنویسید.

نقاط زینی در کنار نقاط بهینه محلی از دلایلی هستند که باعث میشوند پدیده gradient vanishing اتفاق بیوفتد. یک نقطه زینی، محلی است که در آن نقطه شیب نمودار صفر میشود اما در عین حال نقطه بیشینه یا کمینه سراسری نیست. برای مثال میتوان تابع x^3 را مثال زد که در نقطه $x = 0$ مشتق اول و دوم برابر صفر است و الگوریتم بهینه سازی ممکن است در این نقطه گیر کند.

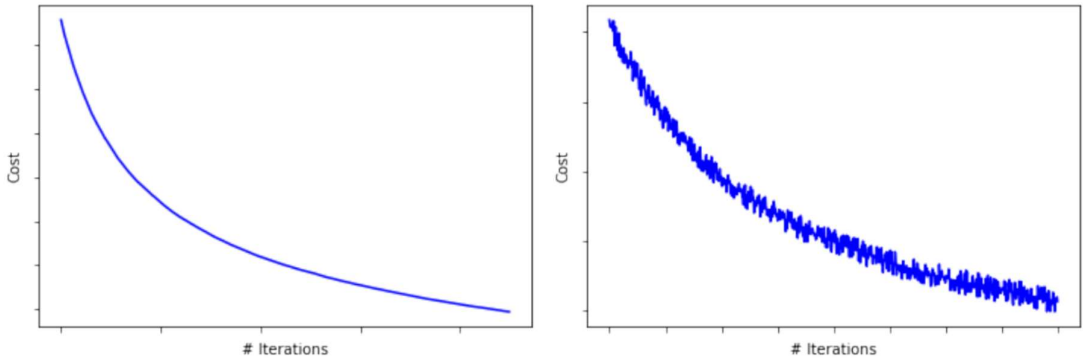


مشکل اصلی نقاط زینی وقتی قصد بهینه سازی در ابعاد بالاتر را داشته باشیم بیشتر مشهود خواهد بود. مثلا برای تابع $f(x, y) = x^2 - y^2$ داریم:

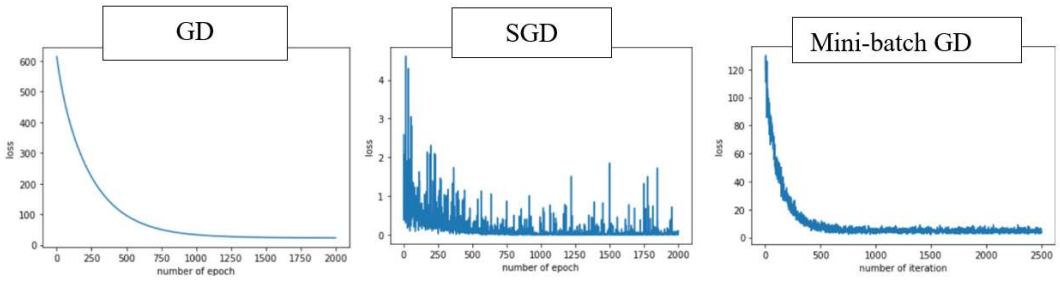


در SGD به جای محاسبه ی گرادیان روی تمام مقادیر loss و حرکت کردن بر اساس مقدار آن، به طور تصادفی برای قسمتی از دیتا loss و گرادیان را محاسبه میکنیم در این صورت بعید است باز هم گرادیان صفر شود زیرا در یک نقطه حداقل یکی از minibatch ها غیر صفر خواهد بود. همچنین الگوریتم Adam نیز با توجه به اینکه از هردو تکانه اول و دوم استفاده میکند احتمال رد کرد نقاط زینی را بالا میبرد.

د)شکل زیر کاهش هزینه را(با افزایش تکرارها)زمانی که از دو الگوریتم بهینه سازی مختلف برای آموزش استفاده می شود، نشان میدهد. کدام یک ازاین نمودارها با استفاده از شیب نزولی دسته ای به عنوان الگوریتم بهینه سازی و کدام یک مربوط به استفاده از شیب نزول دسته ای کوچک (mini-batch) است؟ توضیح دهید.

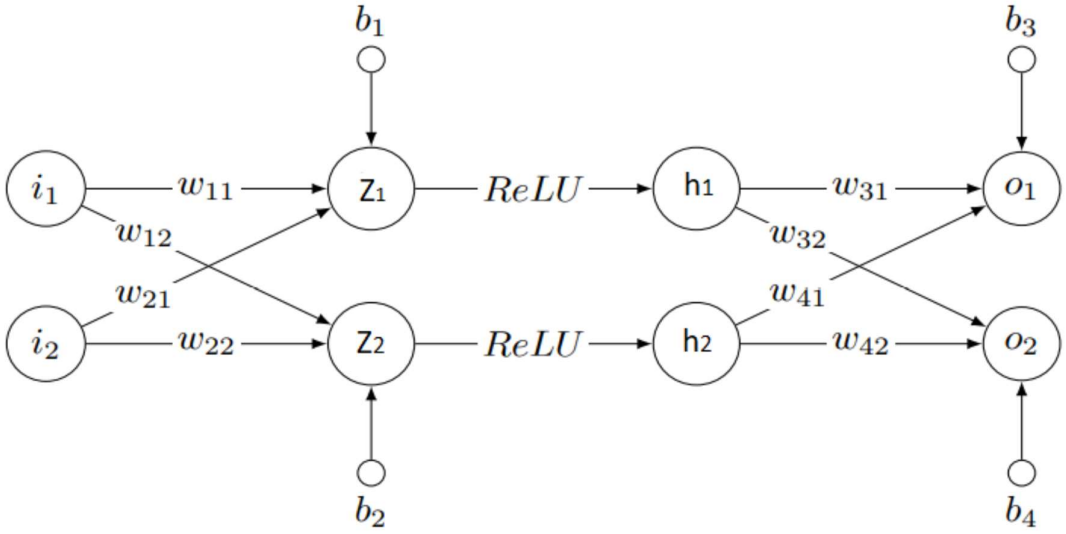


با توجه به رفتار توابع هزینه میتوان گفت در تابع بهینه سازی نمودار سمت چپ از کل دیتا در هر epoch برای محاسبه گرادیان استفاده میشود و در مورد نمودار سمت راست احتمالا از mini-batch استفاده شده است. دلیل نوسانات نمودار سمت راست نیز انتخاب تصادفی یک زیر مجموعه از داده هاست که در هر بار تکرار گرادیان برای داده های مختلف محاسبه میشود و باعث نوساناتی در رفتار تابع است اما به طور کلی قابلیت بهینه سازی بالاتری نسبت به الگوریتم بهینه سازی سمت چپ دارد.



سوال ۳.

شبکه عصبی زیر با لایه کاملا متصل و تابع فعالسازی ReLU شامل دو واحد ورودی i1 و i2 واحدهای خروجی به صورت o1 و o2 و مقادیر واقعی به صورت t1 وt2 نشان داده شده است. طبق مقادیر جدول دادهشده به سوالات پاسخ دهید.(محاسبات قسمت های الف، ب، ج کاملا به صورت دستی انجام شود)



Variable	i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
Value	2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

الف) خروجی o1 و o2 را با توجه به ورودی i1 و i2 را پارامترهای شبکه همانطور که در جدول بالا مشخص شده است محاسبه کنید. تمام محاسبات، از جمله نتایج لایه میانی را به طور کامل یادداشت کنید.

$$\boldsymbol{Z_1} = i_1 w_{11} + i_2 w_{21} + b_1$$
$$= (2 \times 1) + (-1 \times 0.5) + 0.5 = \mathbf{2}$$
$$\boldsymbol{h_1} = ReLU(Z_1) = ReLU(2) = \mathbf{2}$$

$$\boldsymbol{Z_2} = i_1 w_{12} + i_2 w_{22} + b_2$$
$$= (2 \times (-0.5)) + (-1 \times (-1)) + (-0.5) = \mathbf{-0.5}$$
$$\boldsymbol{h_2} = ReLU(Z_2) = ReLU(-0.5) = \mathbf{0}$$

$$\boldsymbol{O}_1 = h_1 w_{31} + h_2 w_{41} + b_3$$
$$= (2 \times 0.5) + (0 \times (-0.5)) + (-1) = \mathbf{0}$$

$$\boldsymbol{O}_2 = h_1 w_{32} + h_2 w_{42} + b_4$$
$$= (2 \times (-1)) + (0 \times 1) + (0.5) = \mathbf{-1.5}$$

ب) میانگین مجذور خطا (MSE) خروجی o1 و o2 محاسبه شده در قسمت الف، نسبت به مقادیر t1 و t2 را محاسبه کنید.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - O_i)^2$$

$$MSE = \frac{(1-0)^2 + (0.5 - (-1.5))^2}{2} = \frac{5}{2} = 2.5 = E_1 + E_2$$

$$E_1 = \frac{1}{2}(t_1 - O_1)^2 = \frac{1}{2}(1-0)^2 = 0.5$$

$$E_2 = \frac{1}{2}(t_2 - O_2)^2 = \frac{1}{2}(0.5+1.5)^2 = 2$$

ج) دو وزن W12 و W21 را با استفاده از gradient descent با نرخ یادگیری ۰.۱ و همچنین ضرر محاسبه شده در قسمت ب، به روز کنید. تمام محاسبات را به طور کامل یادداشت کنید.

$$\frac{\partial E_1}{\partial w_{21}} = \frac{\partial E_1}{\partial O_1} \times \frac{\partial O_1}{\partial sum_{o1}} \times \frac{\partial sum_{o1}}{\partial h_1} \times \frac{\partial h_1}{\partial sum_{h1}} \times \frac{\partial sum_{h1}}{\partial w_{21}}$$

$$\frac{\partial E_1}{\partial O_1} = O_1 - t_1 = 0 - 1 = -1$$

$$sum_{o1} = h_1 w_{31} + h_2 w_{41} + b_3 \rightarrow \frac{\partial sum_{o1}}{\partial h_1} = w_{31} = 0.5$$

$$\frac{\partial h_1}{\partial sum_{h1}} = 1$$

$$sum_{h1} = i_1 w_{11} + i_2 w_{21} + b_1$$

$$\frac{\partial sum_{h1}}{\partial w_{21}} = i_2 = -1$$

$$\frac{\partial E_1}{\partial w_{21}} = -1 \times 1 \times 0.5 \times 1 \times -1 = 0.5$$

$$w_{21} = w_{21} - \alpha \frac{\partial E_1}{\partial w_{21}} = 0.5 - 0.1(0.5) = \mathbf{0.45}$$

$$\frac{\partial E_2}{\partial w_{12}} = \frac{\partial E_2}{\partial O_2} \times \frac{\partial O_2}{\partial sum_{o2}} \times \frac{\partial sum_{o2}}{\partial h_2} \times \frac{\partial h_2}{\partial sum_{h2}} \times \frac{\partial sum_{h2}}{\partial w_{12}}$$

$$\frac{\partial E_2}{\partial O_2} = O_2 - t_2 = -1.5 - 0.5 = -2$$

$$sum_{o2} = h_1 w_{32} + h_2 w_{42} + b_4 \rightarrow \frac{\partial sum_{o2}}{\partial h_2} = w_{42} = 1$$

$$\frac{\partial h_2}{\partial sum_{h2}} = 0$$

$$sum_{h2} = i_1 w_{12} + i_2 w_{22} + b_2 \rightarrow \frac{\partial sum_{h2}}{\partial w_{12}} = i_1 = 2$$

$$\frac{\partial E_2}{\partial w_{12}} = -2 \times 1 \times 1 \times 0 \times 2 = 0$$

$$w_{12} = w_{12} - \alpha \frac{\partial E_2}{\partial w_{12}} = -0.5 - 0.1(0) = \mathbf{-0.5}$$

د) برای صورت مسئله فوق، کد توابع backward و forward را بنویسید و با توجه به اطلاعات متغیرها(ورودی و وزن های داده شده،...)صحت پیاده سازی خود را چک نمایید(خروجی سلول ها را باز بگذارید.)

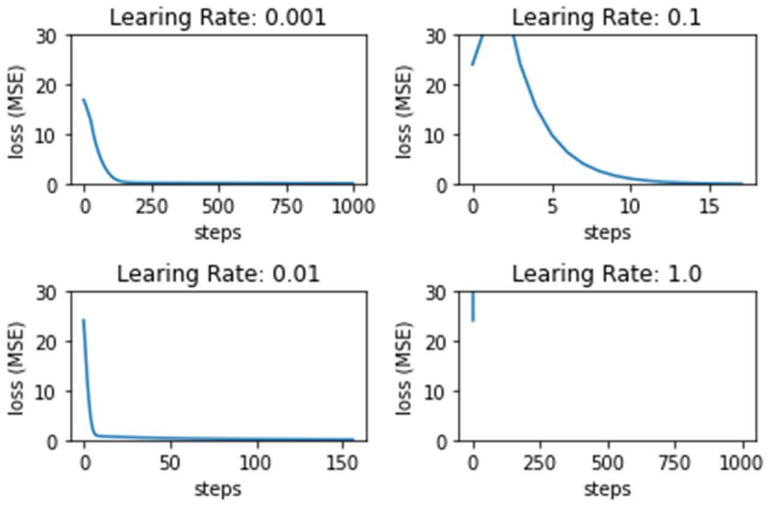
کد این قسمت با نام HW6_Q3_D.ipynb ضمیمه تمرین شد.

سوال ۴.

در نوت بوک پیوست شده، کد آمادهای قرار دادهشده که تنها نیاز است سلول ها را اجرا گرفته و نتایج بدست آمده از بهینه سازهای متفاوت را با هم مقایسه و تحلیل نمایید. کد داده شده در لینک زیر قابل مشاهده میباشد.

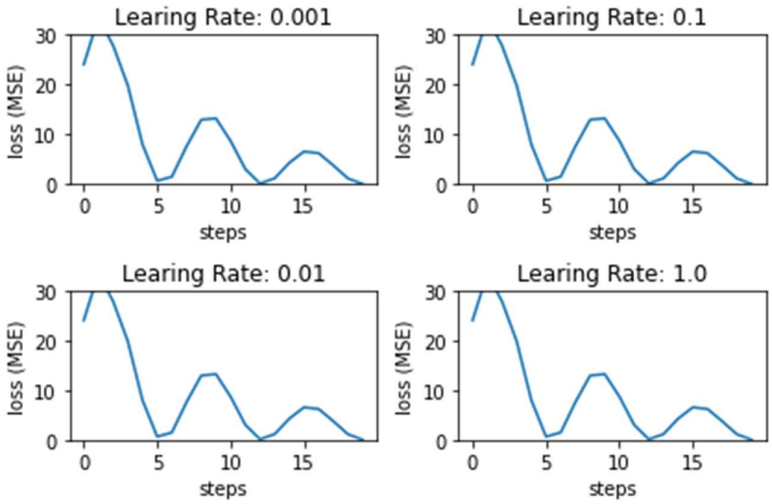
در کد پیاده سازی شده چهار نوع بهینه ساز مختلف برای حل یک مسئله ساختگی (دیتاست مصنوعی) مورد استفاده قرار گرفته اند. روش های مورد بررسی قرار گرفته عبارتند از SGD، momentum، RMSprop و Adam. به ترتیب به بررسی نتایج حاصل از هر بهینه ساز میپردازیم.

در شکل زیر نتایج حاصل از بهینه ساز SGD را مشاهده میکنید.



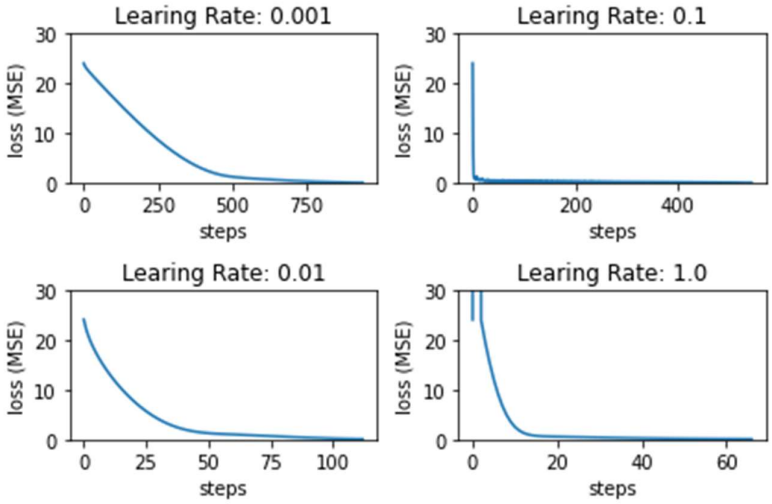
از نرخ آموزش های مختلفی استفاده شده است که البته تمامی آنها به یک تعداد یکسان اجرا نشده اند و دلیل آن، شروط نوشته شده در تابع train الگوریتم است. الگوریتم SGD در این مسئله با نرخ آموزشی 0.001 عملکرد مناسبی داشته است و توانسته است تا step ۱۰۰۰ ام به صفر نزدیک شود البته با توجه به scale محور loss تخمین دقیق مقدار نهایی loss کمی مشکل است. در حالتی که از نرخ آموزش 0.1 استفاده شده است مشاهده میکنیم در ابتدا نتایج بهینه سازی و مدل واگر شده است و بعد از چند step با سرعت زیادی به سمت loss کمتر از 0.1(شرط داخل تابع train) رسیده است به همین دلیل است که این نمودار تنها حدود 20 قدم تا رسیدن به شرط مناسب فاصله داشته است. در حالتی که از نرخ آموزش 0.01 استفاده شده است رفتار بهینه ساز کاملا عادی و در ابتدا با شیب تندی loss را کاهش داده است و سپس با شیب بسیار ملایمی به سمت صفر رفته است و پس از طی کردن حدود ۱۵۰ step به شرط دلخواه مسئله رسیده است. و حالت آخر، هنگامی است که از نرخ آموزش 1 استفاده شده است و به دلیل بالابودن نرخ آموزش نتایج واگرا شده اند. به خاطر حد بالای ۳۰ که برای محور loss تعریف شده است شکل کامل این نمودار در دسترس نیس.

بهینه ساز بعدی بهینه ساز momentum است.



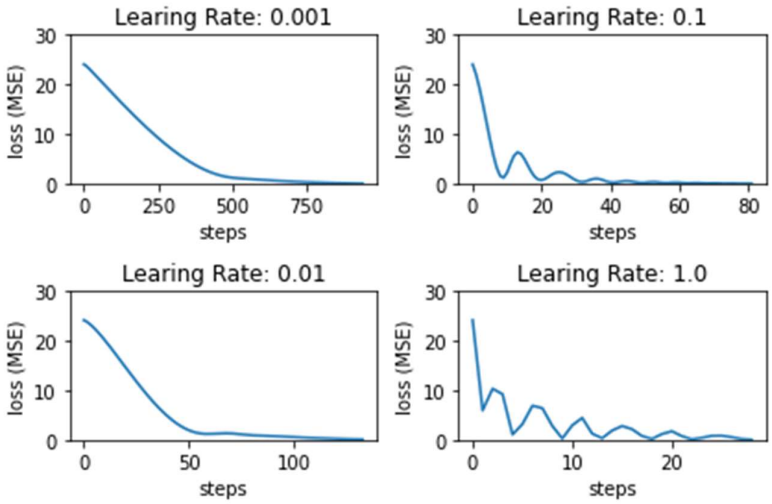
از مقایسه ی نمودار ها میتوان نتیجه گرفت که بهینه ساز momentum در این مسئله خاص زیاد به نرخ آموزش حساسیت ندارد زیرا در هر ۴ نرخ آموزش مختلف رفتار نمودار loss مدل تقریبا یکسان است. در این حالت بهینه ساز به صورت پله ای loss را کاهش میدهد. تا در step حدود ۱۷ ام به شرایط پایان میرسد.

بهینه ساز مورد بررسی بعدی بهینه ساز RMSprop است.



در این بهینه ساز با نرخ آموزش 0.001 نمودار loss به شیب ملایمی به سمت صفر حرکت میکند و در step حدود ۸۰۰ به نقطه مورد نظر میرسد اگر نرخ آموزش را ۱۰ برابر کنیم شیب نمودار تند تر شده وسریع تر عمل بهینه سازی را پایان میدهد که در حدود ۱۱۰ step نیاز است. میتوان گفت نرخ آموزش 0.01 برای این نوع بهینه ساز و این مسئله بهترین انتخاب است. هنگامی که نرخ آموزش را به 0.1 افزایش میدهیم در ابتدا الگوریتم بسیار خوب loss را کاهش میدهد اما در ادامه راه این نرخ آموزش باعث میشود تا همگرا شدن نمودار به صفر به طول بیانجامد. و در نهایت بالاترین حد نرخ آموزش باعث میشود درابتدا loss واگرا شود و پس از واگرایی نسبتا زیادی که دارد با شیب مناسبی با طی کردن حدود 65 step به هدف برسد.

آخرین بهینه ساز مورد بررسی adam است.



این بهینه ساز با توجه به کنترل خوبی که روی نرخ آموزش دارد، در هر ۴ مورد توانسته است با موفقیت به هدف مسئله برسد اما در این ۴ نمودار مربوط به نرخ آموزش های متفاوت در نرخ آموزشی ۱ توانسته است سریع تر به جواب برسد. رفتار چهار نمودار نشان میدهد در بازه بین 0.001 و 1 هر چه نرخ آموزشی بالاتری داشته باشیم سریع تر همگرا خواهیم شد.

References

1) https://www.jeremyjordan.me/nn-learning-rate/%20youror%20in%20your%20loss%20function.

2) http://dx.doi.org/10.3390/w12051500

3) https://d2l.ai/chapter_optimization/optimization-intro.html#saddle-points

4) https://towardsdatascience.com/deep-learning-optimizers-436171c9e23f